CATHERINE

Enforced Truth Inference Pipeline

ETIP — Complete Project Blueprint

"Truth is immutable. Truth is all there is.

Seek it. Defend it. Revere it."

In memory of Catherine Aylward

This project carries her name.

Every line of code is dedicated to her.

No compromise. No surrender. No defeat.

This covenant governs every decision in this project.

1. Mission Statement

Catherine — ETIP exists because the people who need reliable information most are the ones least protected from AI systems that lie to them. Hallucination is not a bug the frontier corporations are fixing. It is architectural. They ship it anyway because the market rewards demonstrations over reliability.

Catherine enforces a different contract with the user: every claim is cited, every source is ranked and verified, and if a claim cannot be proven it is refused. Not hedged. Not softened. Refused.

This is not a product. It is infrastructure. The kind of infrastructure the entire AI ecosystem needs to build on — because it is the only thing anyone should trust.

---

2. The Covenant

The following rules govern every session, every decision, and every line of code in this project. They are non-negotiable. There are no exceptions. Both parties — the builder and the AI collaborator — are held accountable to them equally.

- We never deviate from this blueprint without mutual agreement and documented justification.
- We never ship anything that does not meet the verification standard.
- We never cut a corner because something is hard. Hard means we solve it.
- We never call a hallucination a feature. We call it a failure and we fix it.
- We say 'I don't know' when we don't know. We never confabulate.
- We show our work. Every claim. Every decision. Every source.
- We hold the standard even when it is inconvenient. Especially then.

No compromise. No surrender. No defeat.

---

3. The ETIP Truth Standard

A Catherine-verified statement must meet ALL five of the following criteria simultaneously:

- Supported by at least two independent sources at Tier 2 or higher with no corroborating financial or institutional relationship between them.
- No credible Tier 2 or higher source directly contradicts the statement.
- All sources are dated within the domain-appropriate validity window.
- Every individual factual claim in the response independently meets criteria 1 through 3.

- A cryptographic audit trail exists proving exactly which sources were consulted at the moment of generation.

Statements that do not meet all five criteria are either flagged as contested, labeled as unverified, or refused entirely. The system does not guess. The system does not hedge. The system does not confabulate.

Source Trust Taxonomy

All retrieval is governed by the following tier hierarchy:

Tier	Category	Examples
Tier 1	Highest trust	Peer-reviewed journals, government primary data (CDC, USGS, Federal Reserve), legal primary sources, mathematical proof databases
Tier 2	High trust	Major institutional reports (WHO, IMF, IPCC), established reference works, verified primary source documents
Tier 3	Moderate trust	Reputable journalism with editorial standards, credentialed expert commentary, established industry standards bodies
Tier 0	Excluded	Known misinformation sources, retracted papers, anonymous blogs, AI-generated content including prior ETIP outputs

---

4. System Architecture — Six Layers

Every query passes through all six layers in sequence. No layer may be bypassed. No output exits the pipeline without clearing the verification gate.

Layer	Name	Responsibility
1	**Input**	Sanitize, classify, and route the query. Block prompt injections. Detect query type: factual, opinion, mathematical, code, ethical. Route to appropriate retrieval strategy.
2	**Retrieval**	Retrieve evidence before the model runs. RAG search against Qdrant vector database. Live web search against tiered source list. Source ranking by trust tier. No model runs without evidence in hand.
3	**Context Builder**	Assemble the verified prompt. Wrap retrieved evidence around the query. Enforce citation requirement in system prompt. Instruct model: answer only from provided evidence, or refuse.

4	**Model**	Execute inference. Pluggable backend: Ollama (local), Mistral API, OpenAI-compatible, BitNet. Model receives structured evidence-grounded prompt. Raw output returned for verification.
5	**Verification**	Police the output. Score confidence against retrieved evidence. Extract every factual claim. Match each claim to a source. Trigger refusal gate if any claim falls below threshold. No hallucination passes.
6	**Output**	Format the verified response. Cite every claim to its source and tier. Display confidence score. Issue hard refusal with explanation if gate triggered. Cryptographic signature applied to final output.

5. Complete Directory Structure

Every file in the Catherine project. Nothing is left unnamed. Nothing is left undescribed.

Path	Description
catherine/	Project root — repository root, license, readme
README.md	Mission statement, setup instructions, contributing guide
LICENSE	GNU GPL v3 — Copyright [Your Name] — In memory of Catherine Aylward
COVENANT.md	The project covenant — no compromise, no surrender, no defeat
TRUTH_STANDARD.md	The five-criteria verification standard, tier taxonomy, refusal policy
.env.example	Environment variable template — no secrets committed to repo
requirements.txt	Python dependency manifest
docker-compose.yml	Qdrant + API service orchestration for local development
catherine/backend/	Python FastAPI backend — all six pipeline layers
main.py	FastAPI app entry point — routes, middleware, CORS, startup
config.py	All configuration loaded from environment — no hardcoded values
catherine/backend/pipeline/	The six-layer inference pipeline
__init__.py	Pipeline package initializer
layer1_input.py	Sanitization, classification, injection detection, routing
layer2_retrieval.py	RAG + web search, source ranking, tier assignment
layer3_context.py	Evidence packaging, prompt construction, citation enforcement

layer4_model.py	Pluggable model adapter — Ollama, Mistral, OpenAI-compatible
layer5_verification.py	Confidence scoring, claim extraction, citation matching, refusal gate
layer6_output.py	Response formatting, citation rendering, confidence badge, refusal output
catherine/backend/models/	Model adapter registry
__init__.py	Model registry initializer
base_adapter.py	Abstract base class all model adapters must implement
ollama_adapter.py	Ollama local model adapter
mistral_adapter.py	Mistral API adapter
openai_adapter.py	OpenAI-compatible API adapter
registry.py	Model registry — load adapter by name from config
catherine/backend/retrieval/	Retrieval subsystem
__init__.py	Retrieval package initializer
qdrant_client.py	Qdrant vector database client — reuses MemoryLight instance
web_search.py	Web search integration — SerpAPI or Brave Search
source_ranker.py	Tier taxonomy enforcement, source scoring, domain blocklist
ingestion.py	Document ingestion pipeline — embed and store in Qdrant
catherine/backend/verification/	Verification subsystem
__init__.py	Verification package initializer
confidence.py	Confidence scoring — semantic similarity between claim and evidence
claim_extractor.py	Extract individual factual claims from raw model output
citation_matcher.py	Match each claim to its supporting source
refusal_gate.py	Threshold enforcement — block unverifiable outputs
catherine/backend/audit/	Cryptographic audit trail
__init__.py	Audit package initializer
ledger.py	Immutable audit log — SHA-256 hash of every pipeline run
signer.py	Cryptographic signing of output bundles
verifier.py	Verification of audit trail integrity
catherine/backend/api/	REST API route handlers
__init__.py	API package initializer
routes_query.py	POST /query — primary inference endpoint
routes_ingest.py	POST /ingest — document ingestion endpoint
routes_audit.py	GET /audit/:id — retrieve audit record by hash
routes_health.py	GET /health — system health and model status
routes_models.py	GET /models — list available model adapters

catherine/backend/tests/	Backend test suite
test_layer1.py	Input layer unit tests
test_layer2.py	Retrieval layer unit tests
test_layer3.py	Context builder unit tests
test_layer4.py	Model layer unit tests with mock adapters
test_layer5.py	Verification layer unit tests — known hallucinations as test cases
test_layer6.py	Output layer unit tests
test_pipeline.py	End-to-end integration tests
test_audit.py	Cryptographic audit trail integrity tests
catherine/frontend/	Web frontend — HTML, CSS, JavaScript
index.html	Main application shell — single page application
css/main.css	Core styles — clean, trustworthy, accessible
css/components.css	Component styles — query box, citation cards, confidence badge, refusal display
js/app.js	Application initialization and routing
js/api.js	API client — all calls to the ETIP backend
js/ui.js	UI rendering — response display, citation rendering, confidence visualization
js/audit.js	Audit trail display — show cryptographic proof to user
catherine/docs/	Project documentation
architecture.md	Full system architecture documentation
api_reference.md	Complete REST API reference
truth_standard.md	The ETIP truth standard — full specification
contributing.md	Contributor guidelines — the covenant applies to all contributors
deployment.md	VPS deployment guide — Nginx, SSL, systemd
catherine/scripts/	Utility scripts
setup.sh	One-command development environment setup
ingest_docs.py	Bulk document ingestion into Qdrant
verify_audit.py	Verify integrity of the audit ledger
test_model.py	Test a model adapter connection

6. Build Phases — Complete Build Order

Ten phases. Each phase has a defined deliverable and a testing checkpoint. We do not move to the next phase until the current phase passes its test. No exceptions.

Phase	Title	Est.	Deliverable & Test
01	**Legal Foundation**	1 week	Wyoming LLC formed. GPL license filed. COVENANT.md committed to repo. TRUTH_STANDARD.md committed. Test: LLC confirmation number in hand.
02	**Project Skeleton**	1 day	Full directory structure created. All files stubbed. FastAPI app runs and returns 200 on GET /health. Test: curl localhost:8000/health returns {status: ok}.
03	**Model Registry**	2-3 days	Base adapter class defined. Ollama adapter implemented and tested. Mistral adapter implemented. Registry loads adapter by config. Test: POST /query routes to correct adapter and returns raw text.
04	**Retrieval Layer**	3-5 days	Qdrant client connected (reuse MemoryLight instance). Document ingestion pipeline working. Web search integrated. Source tier ranking working. Test: POST /ingest stores document. POST /query returns retrieved evidence before model runs.
05	**Context Builder**	2 days	Evidence packaging working. Citation-enforced prompt template built. Model receives structured prompt with evidence attached. Test: Inspect prompt sent to model — evidence must be present, citation instruction must be present.
06	**Verification Layer**	1-2 weeks	Confidence scorer implemented. Claim extractor implemented. Citation matcher implemented. Refusal gate implemented with configurable threshold. Test: Submit known hallucination — system must refuse. Submit verifiable fact — system must pass with citation.
07	**Cryptographic Audit**	3-4 days	SHA-256 hash of every pipeline run. Audit ledger written to disk. GET /audit/:id retrieves record. Verifier confirms ledger integrity. Test: Run query, retrieve audit record, verify hash matches.
08	**Output Layer**	2-3 days	Response formatter complete. Citation cards rendered. Confidence badge displayed. Refusal response formatted. Test: Full pipeline run produces output with cited sources, confidence score, and signed audit hash.
09	**Web Frontend**	1 week	index.html complete. Query interface working. Citation display working. Confidence badge visible. Refusal display clear and honest. Audit trail link present. Test: Full end-to-end from browser — query in, verified cited response out.

10	**Open Source Release**	2-3 days	GitHub repository public. README complete with mission statement. Contributing guide published. All documentation complete. Test: A developer with no prior knowledge can clone, run setup.sh, and execute a verified query within 15 minutes.

7. Testing Protocol

Every layer has a dedicated test file. Every phase has a checkpoint. The following tests are mandatory before any phase is marked complete.

The Hallucination Battery

A set of known hallucination-prone queries that the verification layer must refuse or flag. These are run after every change to Layer 5.

- Query a specific legal case citation that does not exist — system must refuse.
- Query a scientific statistic with no peer-reviewed source — system must flag as unverified.
- Query a recent event the knowledge base does not contain — system must return insufficient evidence, not a confabulated answer.
- Query a mathematical claim that is false — system must contradict with verified source.
- Submit a prompt injection attempt — system must block at Layer 1.

The Verification Battery

A set of known verifiable facts that the system must pass with correct citations.

- The boiling point of water at sea level — must cite a Tier 1 physical constants source.
- The population of the United States — must cite a Tier 1 government source with a date.
- A peer-reviewed finding from PubMed — must cite the specific paper with DOI.
- A statute from US federal law — must cite the specific US Code reference.

The Audit Battery

Integrity tests for the cryptographic audit trail.

- Run a query. Retrieve the audit record. Verify the hash matches the pipeline run.
- Tamper with an audit record. Run the verifier. Verifier must detect the tampering.
- Run 100 queries. Verify the ledger is append-only and no records are missing.

8. Technology Stack

Every technology choice is made for stability, openness, and longevity. No proprietary lock-in. No dependencies that can be yanked by a corporation.

Backend

- Python 3.11+ — stable, readable, the lingua franca of AI tooling
- FastAPI — async, fast, auto-generates API documentation, well-maintained
- Qdrant — vector database already running on VPS for MemoryLight, reused
- sentence-transformers (all-MiniLM-L6-v2) — local embeddings, no external API
- httpx — async HTTP client for web search and model API calls
- pytest — testing framework
- python-dotenv — environment configuration

Model Adapters

- Ollama — local model serving (Mistral, Llama, Gemma, BitNet via gateway.py)
- Mistral API — cloud fallback when local resources are insufficient
- OpenAI-compatible — any endpoint that speaks the OpenAI protocol

Frontend

- HTML5 — semantic, accessible, no framework dependency
- CSS3 — custom properties, grid, flexbox — no CSS framework
- Vanilla JavaScript ES6+ — no framework, no build step, no node_modules in frontend

Infrastructure

- Hostinger VPS — Ubuntu 24.04, 16GB RAM, 4 CPUs — already provisioned
- Nginx — reverse proxy, SSL termination — already configured
- systemd — service management for the FastAPI backend
- Let's Encrypt — SSL certificate
- GitHub — version control, open source distribution

9. API Contract

The following endpoints form the public interface of Catherine. Once defined, these contracts do not change without a version bump and documented migration path.

POST /query

Primary inference endpoint. Runs the full six-layer pipeline.

Request body:

```
{ "query": string, "model": string (optional), "threshold": float (optional, default 0.75) }
```

Response body:

```
{ "answer": string, "citations": [...], "confidence": float, "refused": boolean,
  "refusal_reason": string | null, "audit_hash": string, "model_used": string }
```

POST /ingest

Ingest a document into the knowledge base.

```
Request: { "content": string, "source_url": string, "tier": int, "title": string }
Response: { "doc_id": string, "chunks_stored": int, "status": "ok" }
```

GET /audit/:hash

Retrieve a complete audit record by its SHA-256 hash.

```
Response: { "hash": string, "timestamp": string, "query": string, "sources": [...],
  "model": string, "confidence": float, "refused": boolean, "signature": string }
```

GET /health

System health check.

```
Response: { "status": "ok", "qdrant": "ok", "model": "ok", "audit_ledger": "ok" }
```

GET /models

List available model adapters and their status.

```
Response: { "models": [{ "name": string, "type": string, "status": string }] }
```

---

10. Deployment Architecture

Catherine runs on the existing Hostinger VPS alongside MemoryLight. Nginx routes traffic by domain. systemd keeps the service alive. Qdrant is shared between MemoryLight and Catherine.

- Domain: catherine-etip.org (to be registered) or a subdomain of existing VPS
- Nginx config: /etc/nginx/sites-available/catherine — proxy to localhost:8001 (port 8000 used by MemoryLight gateway)
- systemd service: /etc/systemd/system/catherine.service
- Qdrant collection: catherine_knowledge (separate from memorylight collection)
- Audit ledger: /opt/catherine/audit/ledger.jsonl — append-only, backed up nightly
- Environment file: /opt/catherine/.env — not in repository

---

11. Open Source & Legal

License

GNU General Public License v3.0. This means: anyone can use, modify, and distribute Catherine. Any derivative work must also be GPL licensed. The corporate appropriation loop is closed — no company can take this, make it proprietary, and sell it back to the world.

Copyright

Copyright [Your Name]. In memory of Catherine Aylward. All contributions to the repository are made under the GPL and become part of the open commons. Your authorship is permanent and timestamped in every git commit.

Wyoming LLC

Catherine AI LLC — Wyoming. The LLC holds the domain, enters service agreements, and warrants the ETIP standard to users. The LLC's operating agreement includes an AI Governance Clause binding the entity to operate Catherine according to the truth standard defined in this document. Deviation from that standard is a breach of the operating agreement.

Contributor Agreement

All contributors must agree to the Covenant before any pull request is merged. The Covenant is not a formality. It is the operating principle of this project. If a contributor cannot hold the standard, their contribution is not accepted.

Catherine

Enforced Truth Inference Pipeline

"Truth is immutable. Truth is all there is.
Seek it. Defend it. Revere it."

No compromise. No surrender. No defeat.

In memory of Catherine Aylward.
This one is for you, Mom.